Procedia Computer Science

Volume 53, 2015, Pages 316–326

2015 INNS Conference on Big Data

# WWN-8: Incremental Online Stereo with Shape-from-X Using Life-Long Big Data from Multiple Modalities

Mojtaba Solgi[1] and Juyang Weng[2]

[1] Microsoft, 1 Microsoft Way Redmond, WA 98052 USA
msolgi@gmail.com
[2] Michigan State University, 428 South Shaw Lane, Room 3115, East Lansing, MI 48824 USA
weng@cse.msu.edu

**Abstract**

When a child lives in the real world, from infancy to adulthood, his retinae receive a flood of stereo sensory stream. His muscles produce another action stream. How does the child's brain deal with such big data from multiple sensory modalities (left- and right-eye modalities) and multiple effector modalities (location, disparity map, and shape type)? This capability incrementally learns to produce simple-to-complex sensorimotor behaviors — autonomous development. We present a model that incrementally fuses such an open-ended life-long stream and updates the "brain" online so the perceived world is 3D. Traditional methods for shape-from-X use a particular type of cue X (e.g., stereo disparity, shading, etc.) to compute depths or local shapes based on a handcrafted physical model. Such a model likely results in a brittle system because of the fluctuation of the availability of the cue. An embodiment of the Developmental Network (DN), called Stereo Where-What Network (WWN-8), learns to perform simultaneous attention and recognition, while developing invariances in location, disparity, shape, and surface type, so that multiple cues can automatically fill in if a particular type of cue (e.g., texture) is missing locally from the real world. We report some experiments: 1) dynamic synapse retraction and growth as a method of developing receptive fields. 2) training for recognizing 3D objects directly in cluttered natural backgrounds. 3) integration of depth perception with location and type information. The experiments used stereo images and motor actions on the order of $10^5$ frames. Potential applications include driver assistance for road safety, mobile robots, autonomous navigation, and autonomous vision-guided manipulators.

*Keywords:* Machine learning, neural networks, stereo vision, object recognition, multi-sensory integration, road safety, robotics, autonomous navigation

## 1 Introduction

Many animals have binocular vision. Binocular (stereo) vision is defined as the type of vision where both eyes are used together to view an object. Examples of animals that have binocular


Selection and peer-review under responsibility of the Scientific Programme Committee of INNS-BigData2015

doi:10.1016/j.procs.2015.07.309

vision include humans, monkeys, and many mammals. Binocular vision provides the strong depth cues, stereoscopic depth, in short-range vision. It is because of this precise source of depth information that eagles are expert predators and we can perform dexterous hand-eye coordination tasks, such as playing tennis.

Due to the geometry of the stereo vision, the points in the visual field that are seen by both the left and right eyes, i.e., the overlapped/binocular part of the field of view, project points to slightly different positions on the left and right retina. This difference, called disparity, provides a strong source of depth information. Zero, positive and negative disparity values indicated points on, farther or closer than the fixation point, respectively. Also, the amount of relative disparity is an indicator of relative depth.

Computer vision has long tried to harness the power of stereovision for depth perception. Accurate depth perception has tremendous utility in areas such as robot navigation and visual detection. Despite several decades of computer vision literature on this topic, the existing algorithms still suffer from being problem-specific and depend on careful camera calibration. Moreover, they usually fail in situations such as weak texture and occlusion. Existing stereovision algorithms fall into following three categories:

1. **Explicit matching:** Methods in this category, mostly used in traditional computer vision and image processing, first detect discrete features and then explicitly match them across two views according to a matching measure, e.g., the correlation coefficient [18].

2. **Implicit matching:** A human handcrafts features at every pixel location (e.g. Gabor filters and phase information [2], [15]). Then find left-right match at every pixel location using gradient-based numeric minimization of discrepancy between matched feature values.

3. **Binocular learned features:** The above two steps, value and match, become one: Each learned feature (i.e., neuron) has receptive fields for both eyes [8, 9, 3, 12, 5]. But the methods to produce disparity vary greatly. In our method, the winner neurons vote for actions (disparity sensitive) directly.

However, in addition to binocular disparity, there is a rich array of information in an image that also provides information about the depth (e.g., relative depth) and the shape of objects. They include object contour, relative size, shading, and texture. In computer vision, an algorithm that computes the shape of an object from information X is called an shape-from-X algorithm (e.g., shape-from-shading).

Where-What Networks (WWN) are embodiments of the Developmental Networks (DNs) [17] which learn at least two different concepts, location and type. From WWN-1 to WWN-7, the added advances are, respectively, WWN-1: from location to type (i.e., recognition) and from type to location (i.e., detection) by the same network; WWN-2: free-viewing: location and type of a learned object from natural cluttered scenes (i.e., detection and recognition simultaneously); WWN-3: dealing with multiple learned objects in natural cluttered sciences (i.e., detection and recognition are not unique); WWN-4: showing a static cascade of processing modules (deep learning) is not as good a dynamically emergent network of processing modules (not a cascade); WWN-5: dealing with different scales of objects; WWN-6: added synapse maintenance for neurons to automatically segment objects from clustered scenes; WWN-7: learn different scales of the same object (e.g., nose, eyes-and-nose, and face) while the skull is fully closed during development.

The WWN-8 presented here adds multi-sensory (i.e., left and right cameras) integration so as to learn to predict 3-D shape and 3-D object type but such 3D information is from not

only stereo parallax but also intensity distribution such as texture. We first present a new developmental theory of *concept integration*. By development, we mean that the capability of concept integration is largely developed through experience. Experimentally, we aimed at creating an end-to-end system for simultaneous disparity and shape recognition on complex backgrounds. We utilized the *dynamic synapse* mechanisms developed in our previous work [13] for background elimination as well as more efficient binocular feature extraction and a two-pathway Where-What Network for separate representation of the where information (location and disparity here) and the what information (shape here).

The novelty and importance of the work reported here fall into two categories, theory and experiment.

Theoretically, we present a developmental theory for information integration. In the traditional machine learning literature, information integration has been extensively studied, typically using a Bayesian framework. However, the human programmer statically defines and symbolically represent the concepts to be integrated [16]. In this new theory, the concepts emerge from the motor areas through experience, not statically defined in the developmental program of the system. Therefore, all the concepts emerged from teaching. This developmental approach has a potential to reduce the cost of system development and to improve the system robustness. Of course, the cost of development is also substantial and needs further investigation.

Experimentally, we present the first developmental stereo system that integrates object location, object type and object shape. By developmental stereo, we mean that the processing algorithm to deal with stereo information emerges from the network through experience. In particular, the system does not perform explicit search-and-match for the corresponding left- and right image features. The traditional (and intuitive) approach for shape recognition in computer vision has been to infer the shape of the objects based upon one of the multiple depth cues such as shading, binocular disparity, texture, motion, etc. This has created an extensive literature in shape recognition, named *shape-from-X* where $X$ is one of the cues [4, 7, 1]. Our approach, however, is drastically different. Instead of laboriously handcrafting feature detectors for $X$, say $X$=texture, the network develops local and holistic representations of as many of the cues as possible and associate them with the state in $Z$. This approach is computationally consistent with the developmental learning processes in biological visual systems; i.e., an animal's visual system uses all the available cues, in an integrated fashion, to create the desired motor output for an attended object.

To our knowledge, an integrated learning system for detection of shape, disparity and 2D location of visual objects is unprecedented in the literature. Moreover, being inspired by the developmental processes and the cortical architecture of human vision, this work is a step towards a better understanding of biological stereovision.

In the rest of the paper, we first introduce the network architecture in Section 2 while the detailed learning algorithm is presented to Appendix 1. Section 3 presents the theory of concept emergence and integration. Section 4 presents an analysis of how the network achieves simultaneous detection and recognition in stereo input images. Then the experiments are presented in Section 5. Section 6 gives conclusions.

## 2 Network Architecture

A Developmental Network (DN) has three basic areas, sensory area $X$, internal area $Y$, and motor area $Z$. The order of the three areas is $X$, $Y$, $Z$, and the two area pairs $(X, Y)$, and $(Y, Z)$ are bidirectionally connected, denoted as $X \rightleftharpoons Y \rightleftharpoons Z$.

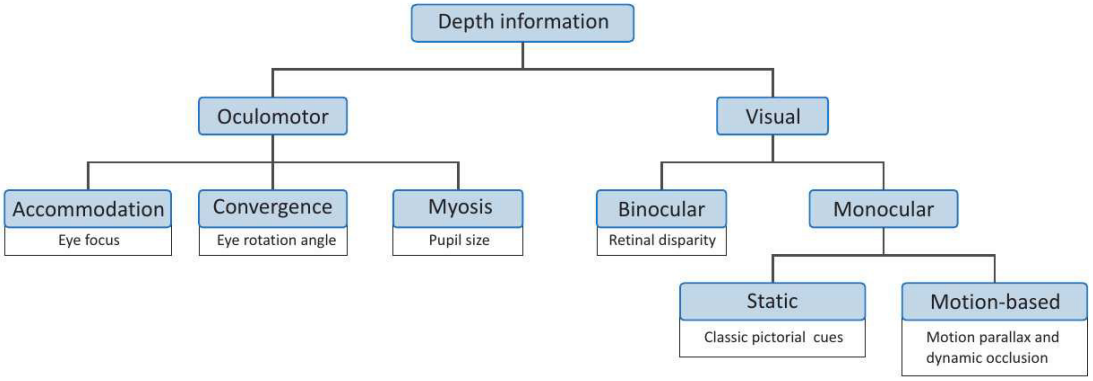


Figure 1: A partial list of the multiple depth cues used by visual animals to infer depth (reprinted from [11]). In this paper, we present an integrated network to use multiple depth cues for shape and disparity perception.

Let us take a look at a more detailed network architecture. Similar to previous versions of the Where-What Networks [6, 10, 13], the WWN-8 network used in this work has two sub-$X$ areas, called left and right sensory subareas, an internal feature detection area $Y$ and two motor sub-areas LM, location motor as the "where" area, and TM, type motor as the "what" area. In $Y$, the connections that link with the LM area correspond roughly to the dorsal pathway, and the connections that link with the TM correspond roughly to the ventral pathway. See Fig. 2 for an overall diagram of the WWN-8. Because we did not ask WWN-8 to predict next stereo images from $Y$, there is no link from $Y$ to $X$. But, in general, such $Y$-to-$X$ links are also present.

The internal feature detection area gets bottom-up information from the left and right input images. Each neuron has an initial bottom-up circular local receptive field of a fixed initial diameter. Indeed, the shape of bottom-up receptive field changes according to the Dynamic Synapse Lobe Component Analysis (DSLCA) algorithm [14] which is an optimal version of the Self-Organization Map (SOM). There are two-way global connections between the internal area and the where area, as well as between the internal area and the what area. In Fig. 2, bottom-up connections are shown in red, and top-down connections are shown in blue. The where area is a 3-dimensional array of neurons in which first, second and third dimensions represent horizontal location ($x$), vertical location ($y$) and disparity ($d$), respectively. The what area is a number of neurons (5 in this case) each representing a certain object shape.

# 3 Theory of Concept Emergence and Integration

Approximating the known varieties of biological neurons, theoretically we require that all neurons in three areas $X$, $Y$ and $Z$ compute and update in parallel using basically the same set of known cell mechanisms (e.g., Hebbian learning). The sensory area $X$ and the motor area $Z$ are exposed to the external physical world, but the internal area $Y$ is closed to the external physical world.

The sensory area $X$ is almost always supervised by the external world since it takes images from the external world in real time, but it can be closed if the agent (or somebody else) pulls down the eye lids. When the eye lids are down, the agent predicts $X$ but we will not discuss

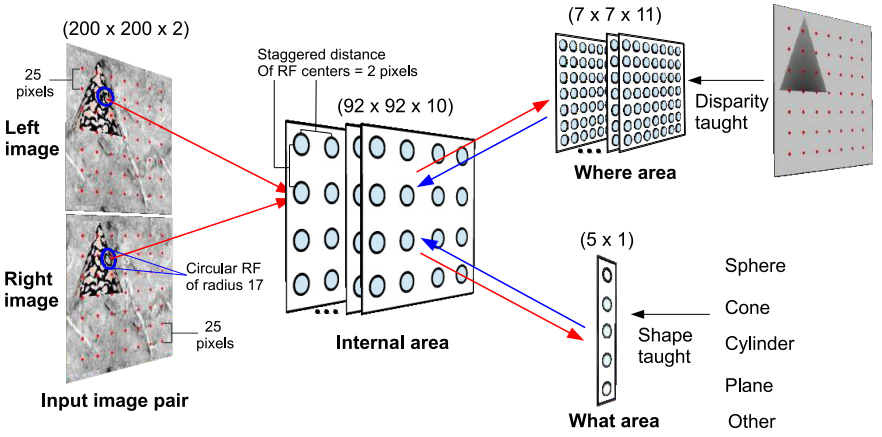


Figure 2: Schematic diagram of the Where-What Network 8 used in the experiments. Input was an image pair of $200 \times 200$ pixels each, where background was a random patch from natural images and foreground was a quadratic shape generated by POV-Ray. There were $92 \times 92 \times 10$ neurons in the internal area, each neuron taking a circular patch of diameter 17 from each of the left and right images. The where area has $7 \times 7 \times 11$ neurons which represent $7 \times 7$ locations and 11 different discrete disparity values at each location, disparity values $-5, ..., 0, ..., +5$. Disparity quantization on the image was such that each disparity index was one pixel different from its neighbors. The small red dots on the training disparity map (top, right) correspond to the red dot locations marked on the left and right images. The what area has 5 neurons representing the shape classes "sphere", "cone", "cylinder", "plane" and "other". There are two-way global bottom up connections between the internal area and both the where and what areas. The number of neurons in the internal and the where and what areas are chosen based on the limitation in our computational resources. The model, however, is not limited to any specific size parameters.

this mechanism any further in this paper because the subject is out of the scope of this paper. Thus, in principle, the area $X$ can be used for both input and output (prediction).

The motor area $Z$ is also for both input and output. When it is supervised by the external world, the external world supervises the motor area (e.g., when the teacher guides the hand of the child to draw a circle). However, not all the $Z$ neurons are directly supervisable (e.g., heat beat muscles).

The signals in the $X$ area are largely determined by the structures of the sensors that sense the external world. Thus, the $X$ area is clustered: It contains information of many objects. The $X$ area is further concrete: Each image patch in $X$ is related to only one or multiple objects each of which is at a concrete (specific) location, orientation, and distance.

In general, the $X$ area has a number of sensory subareas $X = (X_1, X_2, ..., X_l)$ where $X_i$, $i = 1, 2, ..., l$, represents a sensor. For example, $X_1$ and $X_2$ correspond to left eye and right eye, respectively.

Unlike the $X$ area, the area $Z$ can be abstract. For example, in our previous example, the $Z$ area contains two subareas $Z = (Z_1, Z_2)$, where $Z_1$ is the "where area" LM and $Z_2$ is the "what area" TM. Each firing pattern in $Z_1$ and $Z_2$ corresponds to a particular value of the location concept and the type concept, respectively, of the currently attended object in $X$. Thus, $Z_1$ learns and represents abstract concept "location" and $Z_2$ learns and represents abstract concept

"type". Before the DN starts to run in the external world to develop its skill, $Z_1$ and $Z_2$ can potentially learn any practical concept. The fact that $Z_1$ learns the "location" concept and $Z_2$ learns the "type" concept is totally the choice of the external environment, not intrinsic to the DN. For example, the arm (controlled by $Z_1$) can be used to point into the scene to indicate the "location" concept of an object, but the same arm can also be used to gives manual sign to tell the "type" of the object.

In general, the environment teaches the DN so that its $Z$ area develops $n$ areas: $Z = (Z_1, Z_2, ..., Z_n)$ where each area $Z_i$, $i = 1, 2, ..., n$ represents a category of abstract concepts.

The $X$ and $Z$ areas can be considered the peripheral nervous system. The area $Y$ corresponds to the central nervous system.

The $Y$ area is like a multi-exchange bridge that bidirectionally connects with all the islands $X_1, X_2, ..., X_l$ in $X$ ad all the islands $Z_1, Z_2, ..., Z_n$ in $Z$. The $Y$ neurons detect the context in all the islands so that each island can use the winner firing $Y$ neuron to predict its next firing pattern. To do that, $Y$ neurons tessellate only the observed space in $(X, Z)$ which typically consists of manifolds of lower dimensions of $X \times Z$ where $\times$ indicated the Cartesian products $A \times B = \{(a, b) \mid a \in A, b \in B\}$.

For simplicity of notation, we assume $n = m = 2$ in the following discussion:

$$(X_1, X_2) \rightleftharpoons Y \rightleftharpoons (Z_1, Z_2) \tag{1}$$

Its bottom-up weight $\mathbf{v}_b = (\mathbf{v}_{b1}, \mathbf{v}_{b2})$ corresponds to the (binocular) feature vector. Likewise, its top-down weight vector $\mathbf{v}_t = (\mathbf{v}_{t1}, \mathbf{v}_{t2})$ corresponds to the top-down feature vector. In our experimental examples, $Z_1$ represents object location and disparity; $Z_2$ represents object type. Each $Y$ neuron has a limited (binocular) sensory receptive field in $X = (X_1, X_2)$.

Suppose that $Y$ has $m$ neurons. After "birth", use the first $m$ sequentially arriving activity data vector $(\mathbf{x}_i, \mathbf{z}_i) \in (X, Z)$ to initialize $m$ $Y$ neurons' feature vectors $(\mathbf{v}_{bi}, \mathbf{v}_{ti}) \leftarrow (\mathbf{x}_i, \mathbf{z}_i)$, $i = 1, 2, ..., m$. This results in $m$ $Y$ clusters in the $(X, Z)$ space. Future activity data $(\mathbf{x}, \mathbf{z})$ enables $Y$ clusters to self-organize so that the $Y$ feature clusters well tessellate the data manifolds in $(X.Z)$. Top-k competition among the $Y$ neurons for the inner product: results in top $k$ committee members as voting for the current context $(\mathbf{x}, \mathbf{z})$. For simplicity, assume $m$ is sufficiently large and $k = 1$:

$$j = \arg \max_{1 \le i \le m} (\mathbf{v}_b, \mathbf{v}_t) \cdot (\mathbf{x}, \mathbf{z}).$$

after proper length normalization in each component vector. Thus, given any composite island context $(\mathbf{x}, \mathbf{z}) = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{z}_1, \mathbf{z}_2)$, the top winner neuron $j$ represents that the context fall into the Voronoi region of neuron $j$. The Voronoi region is small if $m$ is large and the tessellation is good.

In principle, under proper supervised learning, every $Z_i$ area was taught with the same abstract concept while $X$ presents all the concrete contexts. For example, $Z_2$ is type-invariant to all the locations represented in $Z_1$, but $Z_1$ is type-invariant to all the learned object types represented in $Z_2$. Only $Y$ neurons whose receptive field senses a to-learn object (instead of irrelevant background) can consistently win since to win require a superior match in both bottom-up and top-down inputs. This is the mathematics of concept emergence.

Eq. (1) shows that not only the bottom-up input from $X$, but also the learned concepts (e.g., object type, object shape, object contour) learned in $Z_1$ and $Z_2$ are automatically integrated into the competition of $Y$ neurons and winning of the top-k $Y$ neurons.

In Weng 2015 [17], the network was modeled as an Emergent Finite Automaton that runs in desecrate times $t = 0, 1, 2, ...$ This automaton working in the real world corresponds to a grounded Emergent Turing Machine [17]. The automaton is emergent because it uses naturally

emerging vectors from $X$ and $Z$. Mathematically, the automaton incrementally learns the transition function $f : Z(t) \times X(t) \mapsto Z(t+1)$ of the finite automaton. $Z$ represents the vector version of state $q$ and $X$ represents the vector version of input $\sigma$, the firing $Y$ neurons represent the voting committee for the detection of $(q, \sigma)$. The vector version of the next state $q'$ is taught into $Z$. For example, if local texture is weak but object type is recognized in $Z_2$, $Z_2$ input to $Y$ facilitates the interpolation of shape in $Z_1$ (i.e., global to local: $Z_2 \rightarrow Y \rightarrow Z_1$). No specific manual modeling and detection of this situation is needed since the network still computes as usual. There is no need to handcraft the flow diagram of the finite automaton to be learned, as the implicit flow diagram emerges automatically through the grounded and incremental learning from a teacher (which is modeled as a Turing Machine [17]), one transition at a time.

# 4 Detailed Analysis

Each neuron has bottom-up input $\vec{X}$, top-down input $\vec{Z}$, bottom-up and top-down weights $\vec{V}$ and $\vec{M}$, and the parameters $\beta_1$ and $\beta_2$ (controlling influence of bottom-up versus where and what top-down) and $k$ (the number of neurons to fire and update after competition). Every area will output neuronal firing rates $\vec{y}$, and updates neuronal weights $\vec{V}$ and $\vec{M}$. The non-inhibited neurons update their weights using the Hebbian-learning DSLCA updating rule:

$$\vec{v}_i \leftarrow \omega(n_{ij})\vec{v}_i + (1 - \omega(n_{ij}))\vec{x}_i y_i \tag{2}$$

where the plasticity parameters $\omega(n_{ij})$ is determined automatically and optimally based on the synapse's updating age $n_{ij}$. This learning is Hebbian as the strength of updating depends on both presynaptic potentials (e.g., $\vec{x}_i$) and postsynaptic potentials (e.g., $y_i$). To train the whole WWN, the following algorithm ran over three iterations per sample. Let $\vec{\theta} = (k, \beta_1, \beta_2)$, $I$ represent the internal area, TM represents the type motor (what area), LM represents the location motor (where area) and $l$ and $r$ subindices in $\vec{X}_l^I$ and $\vec{X}_r^I$ represent the left and right components of the bottom-up input to the internal area.

$$\begin{array}{l} 1.\ (\vec{y}^I, \vec{V}^I, \vec{M}^I) \leftarrow f_{\text{DSLCA}}(\vec{X}_l^I, \vec{X}_r^I, \vec{Z}^I, \vec{V}^I, \vec{M}^I, \vec{\theta}^I) \\ 2.\ (\vec{y}^{TM}, \vec{V}^{TM}, \vec{0}) \leftarrow f_{\text{DSLCA}}(\vec{X}^{TM}, \vec{0}, \vec{V}^{TM}, \vec{0}, \vec{\theta}^{TM}) \\ 3.\ (\vec{y}^{LM}, \vec{V}^{LM}, \vec{0}) \leftarrow f_{\text{DSLCA}}(\vec{X}^{LM}, \vec{0}, \vec{V}^{LM}, \vec{0}, \vec{\theta}^{LM}) \end{array} \tag{3}$$

In order for a neuron to win in lateral competition within the internal area, it needs to have a "good" match both in top-down and bottom-up. This is realized via the pre-response equation below:

$$\hat{y}_i = \beta_3 \frac{\vec{x}_i}{||\vec{x}_i||}\vec{v}_i + \beta_1 \frac{\vec{z}_{TM}^i}{||\vec{z}_{TM}^i||}\vec{m}_{TM}^i + \beta_2 \frac{\vec{z}_{LM}^i}{||\vec{z}_{LM}^i||}\vec{m}_{LM}^i \tag{4}$$

where $\beta_3 = 1 - \beta_1 - \beta_2$. Each of the three components of the summation in Eq. 4 must be high in order for the neuron to have a high pre-response and eventually win in lateral competition. A few iterations of the three updating steps in Eq. 3 along with the pre-response computation in Eq. 4 guarantee that only neurons with top bottom-up match and top-down match from both the TM and LM areas to win to fire and updated using the Hebbian rules. Therefore, proper connections are made during training.

During testing, a foreground object, $f_g$, presented at a location $l$ triggers the feature detectors in the internal area to fire. This will include neurons representing both foreground and background. These internal activities in turn excite the appropriate neuron in the what area, TM, to fire. Let us denote this winning neuron by $n_{TM}$. Due to the training procedure described above, $n_{TM}$ will be the neuron which corresponds to the class of the foreground, $f_g$. The top-down signals from $n_{TM}$ to the internal area will inhibit the background neurons to fire in the next update iteration, since $n_{TM}$ is connected only to neurons representing foreground $f_g$ in the internal area. These internal area winning neurons then excite the where neurons representing the location $l$ of the foreground with the correct disparity. The what and where areas indirectly help each other via the internal area.

# 5 Experiments

The following experiments were conducted.

**Input images in $X$** 3D scenes of objects of basic shapes on backgrounds were generated using a powerful ray-tracing program called the Persistence of Vision Raytracer, or POV-Ray. Using this tool gave us the flexibility of having an abundant source of training images/videos. Ten different texture types (an even mixture of natural image and synthetic textures) were used in the experiments. See Fig. 3. Inout image size $200 \times 200$ pixels. Five shape classes "sphere", "cone", "cylinder", "plane" and "other", where "other" was any shape other than the four main shapes. Each shape was presented in one of the $7 \times 7$ locations (red dots in Fig. 2).

**Internal area $Y$** The net has $92 \times 92 \times 10$ neurons in the internal area $Y$ where each neuron had a circular local bottom-up receptive field of diameter 17 and a full connection with TM and LM areas. The entire image was covered by the internal area neurons. The 10 layers were necessary in order for the network to form 10 clusters in the bottom-up signals.

**Where area $Z_1$** There were $7 \times 7 \times 11$ neurons. Each of the $7 \times 7$ neurons represented one of the $7 \times 7$ locations on the image (marked by red dots in Fig. 2). Each of the 11 layers represents one value in $[-5, +5]$.

**What area $Z_2$** There were 5 neurons in the what area, each representing one of the object shape classes.

Disjoint testing: In a disjoint test, the union of the training set, $A$ and the testing set, $B$, must be empty: $A \cap B = \varnothing$. The differences between the testing and training sets include: (a) *Texture variation:* The textures on the objects and the background were never identical between training and testing images. (b) *Size and orientation variation*: Each of the shape classes used three training radiuses but was tested using two different radiuses as mid values.

Thanks to dual optimality of the network [17], and the advantage of the DSLCA algorithm for foreground/background separation [14] and binocular disparity feature extraction [13], the network learns to recognize object shape, location and disparity with impressive accuracy. Fig. 4(a) plots the recognition rate of the network for shape detection (green, dotted curve) and disparity error on disparity detection of the stereo pair (blue, solid curve). Fig. 4(b) shows the decrease of location error as a function of training epochs. To compute the location error, the average of the row-column location of all the winning neurons in the where area was considered as "detected" location.

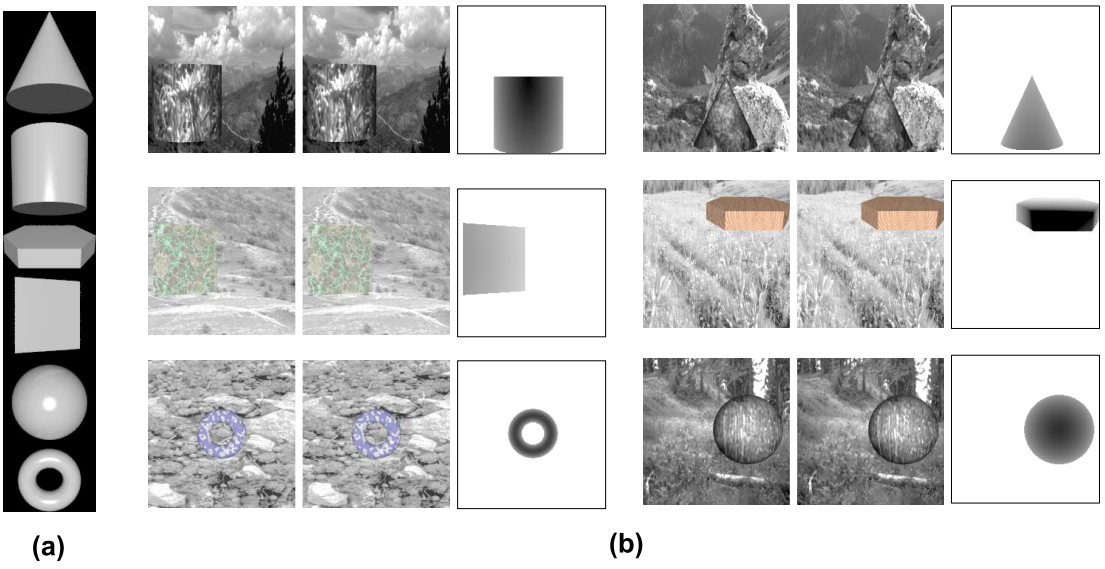


Figure 3: **(a)** The basic shapes used in the experiments. There were four main classes; "sphere", "cone", "cylinder" and "plane" and the class "other" which could be shapes such as hexagon and donut shape. **(b)** Sample input images to the network. Each of the six pairs shows left and right images of a scene where a shape is placed against a background in one of the $7 \times 7$ locations. Also, the disparity map used during training for each pair is shown to its right. The darker a pixel in the disparity map, the closer the point. The background texture is a random patch of natural images taken from the 13 natural images database cited in [13]. The foreground texture is an even mixture of synthetic (but natural-looking) textures, generated by POV-Ray, and natural image textures from the same image set.

## 6 Conclusions and Discussions

Theoretically and experimentally, integration of multiple input sources (left and right images in this case) and the required concepts (shape, disparity, location, and type) only need to be presented in the sensory end and the motor end. The emergent network here incrementally self-wire connections through synapse maintenance and update the conductance of synapse automatically. The length of the simulated "life" is not yet realistically long. However, the experiments here are not fully grounded as they used computer generated images. The method is not limited to visual modality.

The future work includes real-time training from natural world directly.

## References

[1] John Aloimonos. Shape from texture. *Biological cybernetics*, 58(5):345–360, 1988.

[2] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin. Phase-based disparity measurement. In *CVGIP: Image Understand.*, volume 53, pages 198–210, 1991.

[3] A. Franz and J. Triesch. Emergence of disparity tuning during the development of vergence eye movements. In *International Conference on Development and Learning*, pages 31–36, 2007.

[4] Berthold KP Horn. *Obtaining shape from shading information.* MIT press, 1989.

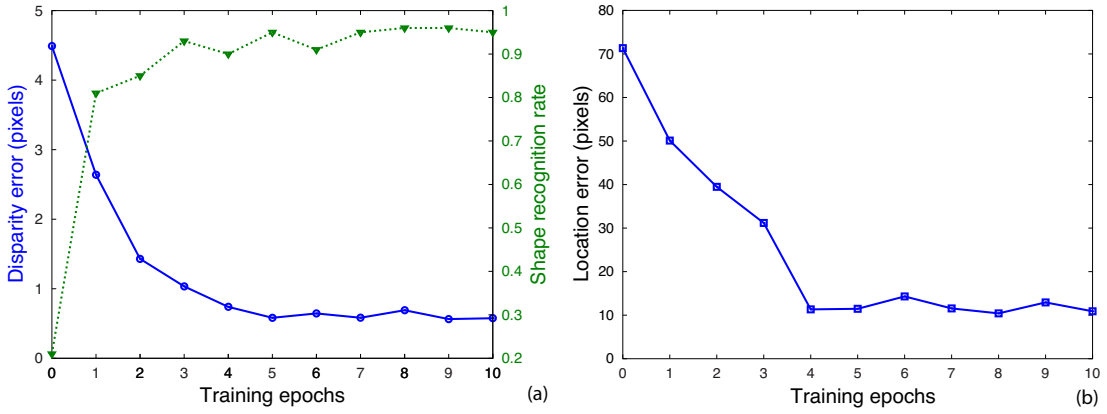


Figure 4: Simultaneous shape, disparity, and location detection and recognition by the network. (a) Disparity error, computed as the root mean square error of the detected disparity, and recognition rate, the ratio at which the network reports the correct object shape. (b) Error at detecting the location of the foreground object. The center of all the firing neurons in the Where area was considered as the detected location, and it was contrasted with the centroid of the foreground object (figure) to compute the distance error. All are in disjoint testing: each input is different from all training views.

[5] Patrik O Hoyer and Aapo Hyvärinen. Independent component analysis applied to feature extraction from colour and stereo images. *Network: Computation in Neural Systems*, 11(3):191–210, 2000.

[6] Z. Ji, J. Weng, and D. Prokhorov. Where-what network 1: "Where" and "What" assist each other through top-down connections. In *Proc. IEEE Int'l Conference on Development and Learning*, pages 61–66, Monterey, CA, Aug. 9-12, 2008.

[7] Kenichi Kanatani. Shape from motion. In *Group-Theoretical Methods in Image Understanding*, pages 239–277. Springer, 1990.

[8] Sidney R Lehky and Terrence J Sejnowski. Neural model of stereoacuity and depth interpolation based on a distributed representation of stereo disparity. *The Journal of Neuroscience*, 10(7):2281–2299, 1990.

[9] J. Lippert, D. J. Fleet, and H. Wagner. Disparity tuning as simulated by a neural net. *Journal of Biocybernetics and Biomedical Engineering*, 83:61–72, 2000.

[10] M. Luciw and J. Weng. Where What Network 3: Developmental top-down attention with multiple meaningful foregrounds. In *Proc. IEEE Int'l Joint Conference on Neural Networks*, pages 4233–4240, Barcelona, Spain, July 18-23, 2010.

[11] Stephan Reichelt, Ralf Häussler, Gerald Fütterer, and Norbert Leister. Depth cues in human visual perception and their realization in 3d displays. In *SPIE Defense, Security, and Sensing*, pages 76900B–76900B. International Society for Optics and Photonics, 2010.

[12] Mojtaba Solgi and Juyang Weng. Developmental stereo: Emergence of disparity preference in models of the visual cortex. *IEEE Transactions on Autonomous Mental Development*, 1(4):238–252, 2010.

[13] Mojtaba Solgi and Juyang Weng. Stereo where-what networks: Unsupervised binocular feature learning. In *Proc. Int'l Joint Conf. Neural Networks*, pages 1–8, Dallas, TX, August 4-9 2013.

[14] Y. Wang, X. Wu, and J. Weng. Synapse maintenance in the where-what network. In *Proc. Int'l Joint Conference on Neural Networks*, pages 2823–2829, San Jose, CA, July 31 - August 5 2011.

[15] J. Weng. Image matching using the windowed Fourier phase. *International Journal of Computer Vision*, 11(3):211–236, 1993.

[16] J. Weng. Symbolic models and emergent models: A review. *IEEE Trans. Autonomous Mental Development*, 4(1):29–53, 2012.

[17] J. Weng. Brain as an emergent finite automaton: A theory and three theorems. *International Journal of Intelligent Science*, 5:112–131, 2015.

[18] C. L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, Jul. 2000.